

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний технічний університет
«Харківський політехнічний інститут»

МЕТОДИЧНІ ВКАЗІВКИ
до лабораторної роботи
«Використання багатовимірних масивів»
з курсу «Програмування»
для студентів напрямку 6.040302 – Інформатика
і курсу «Програмування та алгоритмічні мови»
для студентів напрямку 6.040303 – Системний аналіз

Затверджено редакційно-видавничою
радою університету,
протокол № 3 від 28.12.09.

Харків НТУ «ХПІ» 2010

Методичні вказівки до лабораторної роботи «Використання багатовимірних масивів» з курсу «Програмування» для студентів напрямку 6.040302 – Інформатика і курсу «Програмування та алгоритмічні мови» для студентів напрямку 6.040303 – Системний аналіз / Уклад. М. І. Безменов. – Х. : НТУ «ХП», 2010. – 24 с.

Укладач М. І. Безменов

Рецензент Л. М. Любчик

Кафедра системного аналізу і управління

Мета роботи

Освоєння сфери застосування багатовимірних масивів та методів розв'язання задач із програмування з їх використанням.

1. ТЕОРЕТИЧНІ ОСНОВИ

1.1. Багатовимірні масиви у C++Builder

Дуже часто структуру даних зручно подавати у вигляді прямокутної таблиці. Щоб звернутися до елемента такої структури, потрібно вказати дві координати – по горизонталі і по вертикалі. Для подання таких структур у C++Builder використовують двовимірні масиви. Відмінність в описі двовимірного масиву від одновимірного полягає тільки в тому, що в цьому разі вказуються два розміри, перший з яких людина сприймає як кількість рядків масиву, а другий – як кількість стовпців. Сам по собі двовимірний масив у C++Builder реалізований як масив масивів і має такий формат опису:

```
тип ім'я_масиву [розмір_1] [розмір_2] ;
```

Наприклад, щоб описати двовимірний масив з дійсними елементами, який має 3 рядки та 5 стовпців, можна скористатися таким оператором:

```
double A[3][5] ;
```

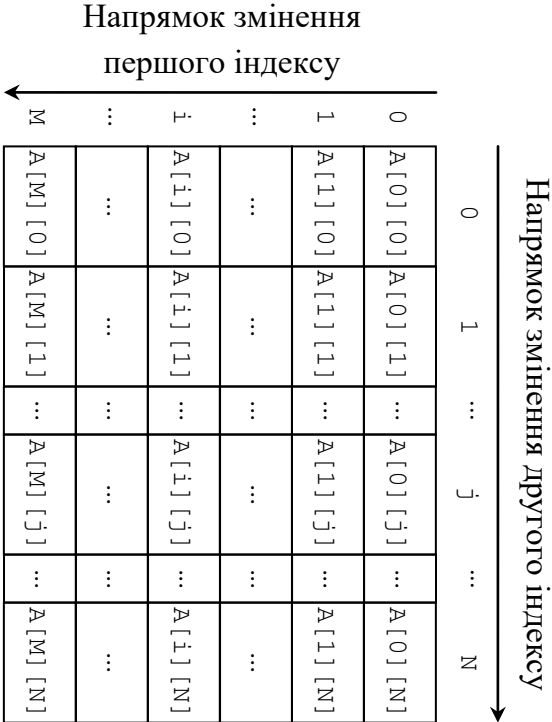
Для звертання до елемента двовимірного масиву необхідно вказати ім'я масиву й значення двох індексів, кожний з яких укладається у **свої квадратні дужки**. Перший індекс указує номер рядка, а другий – номер стовпця, на перетинанні яких розташовано елемент:

```
A[i][2] = 5;
```

Як реалізується двовимірний масив?

Те, що двовимірний масив є масивом масивів, означає, що він є одновимірним масивом, базовим типом якого є також одновимірний масив. Так, описаний вище масив A – це одновимірний триелементний масив, кожний з елементів якого являє собою п'ятиелементний масив – рядок двовимірного масиву. Оскільки елементи одновимірного масиву розміщуються в безперервній області пам'яті в суміжних комірках, рядки двовимірного масиву йдуть у пам'яті один за одним, що відображає рис. 1.1.

Сприйняття двовимірного масиву людиною



Вважається, що масив A має такий опис:
 тип $A[M + 1][N + 1]$;
 де M, N – натуральні числа.

Розміщення елементів двовимірного масиву у пам'яті



Рис. 1.1. Сприйняття двовимірного масиву людиною та розміщення його елементів у пам'яті

У більшості випадків те, що двовимірний масив насправді є одновимірним, не важливо, і можна діяти так, начебто він є масивом із двома індексами.

Як і одновимірний масив, двовимірний масив може бути ініціалізований при описі. Оскільки, як це говорилося раніше, двовимірний масив розглядається як масив масивів, при його ініціалізації вміст кожного з рядків записують у своїх фігурних дужках, вбираючи вміст всього двовимірного масиву у спільні фігурні дужки, розділяючи вміст послідовних рядків комою:

```
int Array2x3[2][3] = {{11, 12, 13}, {21, 22, 23}};
```

Якщо у яких-небудь внутрішніх дужках вказати меншу кількість значень порівняно з кількістю елементів у рядку масива, то здійснюється ініціалізація тільки перших елементів відповідного рядка. Внутрішні фігурні дужки можна опустити. У цьому разі здійснюється ініціалізація тільки перших елементів масиву.

Масиви можуть мати розмірність більшу двох. При цьому в пам'яті вони розташовуються так, що останній індекс змінюється швидше за всі попередні, а перший – повільніше за всі.

Аналогічно описуються типізовані константи – багатовимірні масиви (зовнішні дужки відповідають першому індексу, а дужки найбільшого рівня вкладеності – останньому):

```
double z[2][3][4] = {{{1.11, 1.12, 1.13, 1.14},  
                      {1.21, 1.22, 1.23, 1.24},  
                      {1.31, 1.32, 1.33, 1.34}},  
                     {{2.11, 2.12, 2.13, 2.14},  
                      {2.21, 2.22, 2.23, 2.24},  
                      {2.31, 2.32, 2.33, 2.34}}};
```

Загальний обсяг пам'яті, займаної багатовимірним масивом, не може перевищувати 2 Гбайт. Таке ж саме обмеження має місце і для будь-якої його частини (підмасиву), що відповідає довільному з вимірів.

1.2. Багатовимірні масиви у Delphi

Дуже часто структуру даних зручно подавати у вигляді прямокутної таблиці. Щоб звернутися до елемента такої структури, потрібно вказати дві координати – по горизонталі і по вертикалі. Для подання таких структур у

Delphi використовують двовимірні масиви. Відмінність в описі двовимірного масиву від одновимірного полягає тільки в тому, що необхідно зазначити верхню і нижню межі другого індексу.

```
var  
    ім'я: array [нижня_межа_1..верхня_межа_1,  
                нижня_межа_2..верхня_межа_2] of тип_елементів;
```

Можна застосовувати змішане індексування (наприклад, перший індекс має цілий тип, а другий – символьний).

Крім звичайного способу опису двовимірного масиву, можливий також розгляд його при описі як масиву масивів. Наприклад, такі два описи є ідентичними:

```
var  
    eval: array[1..100, 1..4] of Byte;  
var  
    eval: array[1..100] of array[1..4] of Byte;
```

При використанні другого варіанта опису краще спочатку визначити деякий тип одновимірного масиву (рядок двовимірного масиву), що потім буде використаний при описі двовимірного масиву:

```
type  
    exam = array[1..4] of Byte;  
var  
    eval: array[1..100] of exam;
```

Для звертання до елемента двовимірного масиву необхідно вказати ім'я масиву й у квадратних дужках через кому – значення двох індексів: перший вказує номер рядка, а другий – номер стовпця, на перетинанні яких розташовано елемент (`eval[i, 4] := 5`). Індокси можуть укладатися кожний у свої квадратні дужки без відділення комою (`eval[1][4] := 5`). Такий метод звертання підкреслює те, що насправді двовимірний масив є масивом масивів.

Як реалізується двовимірний масив?

Те, що двовимірний масив є масивом масивів, означає, що він є одновимірним масивом, базовим типом якого є також одновимірний масив. Так, опис

var

A: array[0..2, 0..4] **of** Real;

визначає, що масив A – це одновимірний триелементний масив, кожним з елементів якого якого являє собою п'ятиелементний масив – рядок двовимірного масиву. Оскільки елементи одновимірного масиву розміщуються в безперервній області пам'яті в суміжних комірках, рядки двовимірного масиву йдуть у пам'яті один за одним, що відображає рис. 1.1, у припущенні що масив A описаний так:

A: array [0..M, 0..N] **of** тип;

У більшості випадків те, що двовимірний масив насправді є одновимірним, не важливо, і можна діяти так, начебто він є масивом із двома індексами.

Відзначимо, що у разі еквівалентності типів двох масивів (поза залежністю від кількості вимірів) у Delphi дозволене присвоювання одного масиву іншому.

Наприклад, нехай мають місце такі описи:

type T1 = array[1..4] **of** Real;

var

v, w: array[1..100] **of** T1;

z: T1;

Тоді оператор **v := w** переписує всі елементи масиву **w** у відповідні елементи масиву **v**.

Якщо при звертанні до двовимірного масиву зазначено тільки один індекс, то отримана конструкція є вказівником на рядок, номер якого визначається значенням індексу (тобто це «ідентифікатор» одновимірного масиву, що є рядком двовимірного масиву). Тому для наведених вище описів допустимими є й такі оператори:

```
v[5] := w[4];           //Переписування рядка у рядок
z := w[13];             //Переписування рядка двовимірного
v[10] := z; //масиву в одновимірний масив і навпаки
```

Масиви можуть мати розмірність більшу двох. При цьому в пам'яті вони розташовуються так, що останній індекс змінюється швидше за всі попередні, а перший – повільніше за всі.

Якщо описується двовимірний масив як типізована константа, то при задаванні значень його елементів він розглядається як масив масивів. При цьому в спільних круглих дужках перераховуються укладені в круглі дужки значення елементів рядків (кожен рядок у своїх дужках):

type

```
T2Arr = array[-1..2, 'A'..'C'] of Real;
```

const

```
a: T2Arr = ((-1.1, -1.2, -1.3), (0.1, 0.2, 0.3),  
            (1.1, 1.2, 1.3), (2.1, 2.2, 2.3));
```

Аналогічно описуються типізовані константи – багатовимірні масиви (зовнішні дужки відповідають першому індексу, а дужки найбільшого рівня вкладеності – останньому):

const

```
z: array[1..2, 1..3, 1..4] of Real =  
    (((1.11, 1.12, 1.13, 1.14),  
      (1.21, 1.22, 1.23, 1.24),  
      (1.31, 1.32, 1.33, 1.34)),  
    ((2.11, 2.12, 2.13, 2.14),  
      (2.21, 2.22, 2.23, 2.24),  
      (2.31, 2.32, 2.33, 2.34))));
```

Діапазон зміни будь-якого індексу не може перевищувати 2 Гбайт. Таке ж саме обмеження має місце і для загального обсягу пам'яті, займаної частиною багатовимірного масиву (підмасиву), що відповідає довільному з його вимірів.

2. ПРИКЛАДИ ПРОГРАМ

Приклад 1. Дано квадратний масив дійсних чисел розміром не більше 30×30. Чи правда, що мінімальний з елементів, розташованих над головною діагоналлю і на ній, перебуває правіше і нижче від максимального елемента масиву? Головна діагональ прямує з лівого верхнього кута в правий нижній.

Використання консольного застосування C++Builder:

```
#include <iostream.h>  
#include <math.h>
```



```

int main()
{
    double a[30][30];
    int i, j,
    n,                                     // Розмірність масиву
    StrMax, ColMax, // Номери рядків і стовпців максимального
    StrMin, ColMin;                       // і мінімального елементів
    cout << "n = ";                       // Задаємо розмірність масиву
    cin >> n;
    cin.get();
    randomize();
    for (i = 0; i < n; i++) // Елементи масиву - випадкові
        for (j = 0; j < n; j++) // дійсні числа від -50.0 до 50.0
            a[i][j] = ((random(10000) / 10000.0) - 0.5) * 100;
    cout << "The initial data:\n";
    cout << "n = " << n << endl;
    for (i = 0; i < n; i++) { // Виводимо рядками масив
        for (j = 0; j < n; j++) // Його елементи розділяємо
            cout << a[i][j] << '\t'; // символом табуляції #9
        cout << endl;
    }
    StrMax = 0; // Вважаємо, що шукані елементи
    ColMax = 0; // збігаються
    StrMin = 0; // і знаходяться у лівому
    ColMin = 0; // верхньому куті масиву
    // Далі у подвійному циклі визначаємо дійсне
    for (i = 0; i < n; i++) // місце розташування
        for (j = 0; j < n; j++) { // шуканих елементів
            if (a[i][j] > a[StrMax][ColMax]) {
                StrMax = i;
                ColMax = j;
            }
            if ((i <= j) // Елемент над головною діагоналлю?
                && (a[i][j] < a[StrMin][ColMin])) {
                StrMin = i;
                ColMin = j;
            }
        }
    }
    if ((ColMin > ColMax) && (StrMin > StrMax)) cout << "Yes";
}

```

```

else cout << "No";
cin.get();
return 0;
}

```

Використання Delphi:

Розмістимо на формі однорядковий редактор TEdit (надамо йому ім'я edInput1) для введення розмірності масиву, багаторядковий редактор TMemo (надамо йому ім'я mmOutput1) для виведення результату та кнопку Button1.

Для розв'язання задачі скористаємося таким опрацювачем події OnClick компонента Button1:

```

procedure TForm1.Button1Click(Sender: TObject);
var
    n: Integer;                //Кількість елементів масиву
    a: array [1..30, 1..30] of Real;           //Масив
    i, j: Integer;
    StrMax, ColMax,           // Номери рядків і стовпців максимального
    StrMin, ColMin: Integer;   // і мінімального елементів
    s: string;
begin
    n := StrToInt(edInput1.Text); // Задаємо розмірність масиву
    Randomize;
    for i := 1 to n do          // Елементи масиву - випадкові
        for j := 1 to n do      // дійсні числа від -50.0 до 50.0
            a[i, j] := (Random - 0.5) * 100;
    mmOutput1.Lines.Add('Початкові дані:');
    mmOutput1.Lines.Add('n = ' + IntToStr(n));
    for i := 1 to n do begin    // Виводимо рядками масив
        s := '';                // Нижче формуємо рядок, що буде виводитись
        for j := 1 to n do     // Його елементи розділяємо символом
            s := s + FloatToStrF(a[i, j], ffFixed, 10, 3) + #9;
        mmOutput1.Lines.Add(s); //табуляції #9
    end;
    StrMax := 1;                // Вважаємо, що шукані елементи
    ColMax := 1;                // збігаються
    StrMin := 1;                // і знаходяться у лівому
    ColMin := 1;                // верхньому куті масиву
    // Далі у подвійному циклі визначаємо дійсне місце

```

```

for i := 1 to n do           // розташування шуканих елементів
  for j := 1 to n do begin
    if a[i, j] > a[StrMax, ColMax] then begin
      StrMax := i;
      ColMax := j;
    end;
    if (i <= j)                // Елемент над головною діагоналлю?
      and (a[i, j] < a[StrMin, ColMin]) then begin
      StrMin := i;
      ColMin := j;
    end;
  end;
if (ColMin > ColMax) and (StrMin > StrMax)
  then mmOutput1.Lines.Add('Правда')
  else mmOutput1.Lines.Add('Неправда');
end;

```

Оскільки програма супроводжується коментарями, зупинимося тільки на фрагменті, призначеному для виведення вмісту двовимірного масиву. Вміст масиву виводиться в подвійному циклі, причому внутрішній цикл служить для формування текстового рядка для виведення. Цей рядок містить елементи рядка масиву. Рядок, що буде виводитись, формується у внутрішньому циклі шляхом зчеплення рядків, одержуваних перетворенням до рядкового виду елементів рядка масиву. Щоб елементи масиву розташовувалися в окремих колонках, між ними вставляється символ табуляції, який має десятковий код 9. Зазначимо також, що для перетворення дійсних чисел до рядкового подання застосовано процедуру `FloatToStrF`, яка дозволяє одержувати різне рядкове подання дійсних чисел. У даному випадку указано, що використовується формат запису з фіксованою точкою (другий параметр дорівнює `ffFixed`) з не більш ніж 10 цифрами в запису числа (третій параметр), з них три – у дробовій частині (четвертий параметр).

Приклад 2. Дано натуральні числа m, n, v, w ($m, n \leq 100; v, w \leq n$) та масив дійсних чисел розміром $m \times n$. Поміняти місцями стовпці з номерами v, w .

Використання консольного застосування C++Builder:

```

#include <iostream.h>
#include <math.h>

```

```

int main()
{
    double a[30][30];
    int i, j,
    m, n,                                // Розмірність масиву
    v, w;                                // Номери стовпців, вміст яких обмінюється
    double buf;                          // Допоміжна змінна
    cout << "m = ";                      // Задаємо кількість рядків масиву
    cin >> m;
    cout << "n = ";                      // Задаємо кількість стовпців масиву
    cin >> n;
    cout << "v = ";                      // Задаємо номери стовпців,
    cin >> v;                            // вміст яких обмінюється
    cout << "w = ";
    cin >> w;
    v--; w--;                            // Перехід до нумерації від 0
    cin.get();
    randomize();
    for (i = 0; i < m; i++)              // Елементи масиву - випадкові
        for (j = 0; j < n; j++) // дійсні числа від -50.0 до 50.0
            a[i][j] = ((random(10000) / 10000.0) - 0.5) * 100;
    cout << "The initial data:\n";
    cout << "n = " << n << endl;
    for (i = 0; i < m; i++) {            // Виводимо рядками масив
        for (j = 0; j < n; j++)          // Його елементи розділяємо
            cout << a[i][j] << '\t';    // символом табуляції #9
        cout << endl;
    }
    for (i = 0; i < m; i++) {
        buf = a[i][v];
        a[i][v] = a[i][w];
        a[i][w] = buf;
    }
    cout << "\nResult:\n";
    for (i = 0; i < m; i++) {            // Виводимо рядками масив
        for (j = 0; j < n; j++)          // Його елементи розділяємо
            cout << a[i][j] << '\t';    // символом табуляції #9
        cout << endl;
    }
}

```

```

cin.get();
return 0;
}

```

Використання Delphi:

Розмістимо на формі однорядковий редактор Edit1 для введення чисел m, n, v, w , мітку Label1, записавши у її властивість Caption текст $m =$, багаторядковий редактор Memo1 для виведення результату та кнопку Button1. Опишемо також у секції **implementation** такі змінні:

```

a: array[1..100, 1..100] of Real;
m, n,                               // Розмірність масиву
v, w: Integer;                       // Номери обмінюваних стовпців

```

Для розв'язання задачі скористаємося таким опрацьовувачем події OnClick компонента Button1:

```

procedure TForm1.Button1Click(Sender: TObject);
var
    i, j: Integer;
    buf: Real;
    s: string;
begin
    case Tag of
        0: begin
            m := StrToInt(Edit1.Text);
            Label1.Caption := 'n =';
            end;
        1: begin
            n := StrToInt(Edit1.Text);
            Label1.Caption := 'u =';
            end;
        2: begin
            v := StrToInt(Edit1.Text);
            Label1.Caption := 'w =';
            end;
    else
        begin
            w := StrToInt(Edit1.Text);
            Label1.Visible := False;

```

```

Edit1.Visible := False;
Button1.Visible := False;
for i := 1 to n do      // Елементи масиву - випадкові
    for j := 1 to n do // дійсні числа від -50.0 до 50.0
        a[i, j] := (Random - 0.5) * 100;
Memo1.Lines.Add('Початкові дані:');
for i := 1 to n do begin    // Виводимо рядками масив
    s := ''; // Нижче формуємо рядок, що виводитиметься
    for j := 1 to n do      // Його елементи розділяємо
        s := s + FloatToStrF(a[i, j], ffFixed, 10, 3) + #9;
    Memo1.Lines.Add(s);      // символом табуляції #9
end;
for i := 1 to n do begin
    buf := a[i, v];
    a[i, v] := a[i][w];
    a[i, w] := buf;
end;
Memo1.Lines.Add('Результат:');
for i := 1 to n do begin
    s := '';
    for j := 1 to n do
        s := s + FloatToStrF(a[i, j], ffFixed, 10, 3) + #9;
    Memo1.Lines.Add(s);
end;
end;
Tag := Tag + 1;
end;

```

Зазначимо, що вище використовувалася властивість Tag форми, яка має тип Integer і умовчуване значення 0. Ця властивість може використовуватися програмістом за власним розсудом. В цьому випадку вона використана для вказівки того, яка зі значень m , n , v , w уводиться.

Приклад 3. Дано натуральне число n ($n \leq 100$) та квадратний масив дійсних чисел розміром $n \times n$. Обернути відносно горизонтальної осі вміст частини масиву між його двома діагоналями, включаючи фрагменти діагоналей.

Використання консольного застосування C++Builder:

```
#include <iostream.h>
#include <math.h>
int main()
{
    double a[100][100];
    int i, j,
    n;                                     // Розмірність масиву
    double buf;                           // Допоміжна змінна
    cout << "n = ";                       // Задаємо розмірність масиву
    cin >> n;
    cin.get();
    randomize();
    for (i = 0; i < n; i++)                // Елементи масиву - випадкові
        for (j = 0; j < n; j++) // дійсні числа від -50.0 до 50.0
            a[i][j] = ((random(10000) / 10000.0) - 0.5) * 100;
    cout << "The initial data:\n";
    cout << "n = " << n << endl;
    for (i = 0; i < n; i++) {              // Виводимо рядками масив
        for (j = 0; j < n; j++)            // Його елементи розділяємо
            cout << a[i][j] << '\t';       // символом табуляції #9
        cout << endl;
    }
    for (j = 0; j < n / 2; j++)            // Йдемо до середини масиву
        for (i = j; i < n - j; i++) {     // Беремо частину стовпця
            buf = a[i][j];
            a[i][j] = a[i][n - 1 - j];
            a[i][n - 1 - j] = buf;
        }
    cout << "\nResult:\n";
    for (i = 0; i < n; i++) {              // Виводимо рядками масив
        for (j = 0; j < n; j++)            // Його елементи розділяємо
            cout << a[i][j] << '\t';       // символом табуляції #9
        cout << endl;
    }
    cin.get();
    return 0;
}
```

Використання Delphi:

Розмістимо на формі однорядковий редактор Edit1 для введення розмірності масиву, багаторядковий редактор Memo1 для виведення результату та кнопку Button1.

Для розв'язання задачі скористаємося таким опрацьовувачем події OnClick компонента Button1:

```
procedure TForm1.Button1Click(Sender: TObject);  
var  
    a: array[1..100, 1..100] of Real;  
    n,                                     // Розмірність масиву  
    i, j: Integer;  
    buf: Real;  
    s: string;  
begin  
    n := StrToInt(Edit1.Text);  
    for i := 1 to n do                     // Елементи масиву - випадкові  
        for j := 1 to n do                 // дійсні числа від -50.0 до 50.0  
            a[i, j] := (Random - 0.5) * 100;  
    Memo1.Lines.Add('Початкові дані:');  
    for i := 1 to n do begin                 // Виводимо рядками масив  
        s := '';                            // Нижче формуємо рядок, що виводитиметься  
        for j := 1 to n do                 // Його елементи розділяємо  
            s := s + FloatToStrF(a[i, j], ffFixed, 10, 3) + #9;  
        Memo1.Lines.Add(s);                 // символом табуляції #9  
    end;  
    for j := 1 to n div 2 do                 // Йдемо до середини масиву  
                                           // Беремо частину стовпця  
        for i := j to n - j + 1 do begin  
            buf := a[i][j];  
            a[i][j] := a[i][n - j + 1];  
            a[i][n - j + 1] := buf;  
        end;  
    Memo1.Lines.Add('Результат:');  
    for i := 1 to n do begin  
        s := '';  
        for j := 1 to n do  
            s := s + FloatToStrF(a[i, j], ffFixed, 10, 3) + #9;  
        Memo1.Lines.Add(s);
```


end;
end;

3. ЗАВДАННЯ НА ЛАБОРАТОРНУ РОБОТУ

За час, відведений для виконання лабораторної роботи (2 академічні години), студент повинен:

1. Розробити алгоритм розв'язання задачі, запропонованої для програмування.
2. Здійснити програмну реалізацію розробленого алгоритму.
3. Здійснити відлагодження програми, виправивши синтаксичні та логічні помилки.
4. Підібрати тестові дані для перевірки програми, включаючи виняткові випадки.
5. Оформити звіт до лабораторної роботи.
6. Відповісти на контрольні запитання.
7. Здати викладачу працездатну програму з демонстрацією її роботи на декількох варіантах вихідних даних.

4. КОНТРОЛЬНІ ЗАПИТАННЯ

1. Яку структуру даних імітує двовимірний масив?
2. Як описується двовимірний масив та масив з більшою кількістю вимірів?
3. Як задається діапазон змінення індексів у описі багатовимірних масивів?
4. Чи можна у описі багатовимірних масивів використовувати змінні?
5. Чи є правильним твердження, що у рядках двовимірного масива вміщуються елементи тільки одного типу?
6. Чи можна у різні рядки двовимірного масива записати дані відмінних один від одного типів?
7. Як здійснюється звертання до елементів багатовимірного масиву?
8. Як розміщуються у пам'яті елементи двовимірного масива?
9. Опишіть реалізацію розміщення у пам'яті елементів масивів з кількістю вимірів, більшою двох.
10. Чи можна при звертанні до елементів багатовимірного масиву використовувати вирази?

11. Як здійснюється ініціалізація багатовимірного масиву у C++Builder (визначення типізованої константи – багатовимірного масива у Delphi).
12. Чи можна при ініціалізації багатовимірного масиву у C++Builder (визначенні типізованої константи – багатовимірного масиву у Delphi) вказувати значення не всіх елементів? Зробіть пояснення.
13. Яке обмеження існує для обсягу пам'яті, що може бути відведена під багатовимірний масив?
14. Нижче наведений фрагмент тексту програми (розглядати відповідну мову програмування).

Мова програмування	
C++Builder	Delphi
<pre>double a[3][4]; int i, j; for (j = 0; j < 4; j++) for (i = 0; i < 3; i++) a[i][j] = (i + 1) * 10 + j + 1;</pre>	<pre>var a: array[1..3, 1..4] of Real; i, j: Integer; // Продовження тексту for j := 1 to 4 do for i := 1 to 3 do a[i][j] := i * 10 + j;</pre>

Чи є помилки у наведеному вище тексті? Зробіть необхідні пояснення.

15. Нижче наведений фрагмент тексту програми (розглядати відповідну мову програмування).

Мова програмування	
C++Builder	Delphi
<pre>double a[3][4]; int i, j; for (j = 0; j < 4; j++) for (i = 0; i < 3; i++) a[j][i] = (i + 1) * 10 + j + 1;</pre>	<pre>var a: array[1..3, 1..4] of Real; i, j: Integer; // Продовження тексту for j := 1 to 4 do for i := 1 to 3 do a[j][i] := i * 10 + j;</pre>

Чи є помилки у наведеному вище тексті? Зробіть необхідні пояснення.

16. Знайдіть помилки (якщо вони є) у наведених нижче оголошеннях масивів і зробіть необхідні пояснення (розглядати відповідну мову програмування).

Мова програмування	
C++Builder	Delphi
<code>int x[2][3] = {{-2, -1}, {0, 1}, {2, 3}};</code>	<code>const x: array[1..2, 1..3] of Integer = ((-2, -1), (0, 1), (2, 3));</code>
<code>int x[2][3] = {{-2, -1, 0}, {1, 2, 3}};</code>	<code>const x: array[1..2, 1..3] of Integer = ((-2, -1, 0), (1, 2, 3));</code>
<code>int x[2][3] = { {-2, -1, 0, 1}, {2, 3}};</code>	<code>const x: array[1..2, 1..3] of Integer = ((-2, -1, 0, 1), (2, 3));</code>
<code>int x[2][3] = {-2, -1, 0, 1, 2, 3};</code>	<code>const x: array[1..2, 1..3] of Integer = (-2, -1, 0, 1, 2, 3);</code>
<code>const int SIZE = 4; int x[2 * SIZE][SIZE + 4];</code>	<code>const SIZE = 4; x: array[1..2 * SIZE, 1..SIZE + 4] of Integer;</code>
<code>const int SIZE = 4; int x[2 * SIZE][SIZE - 4];</code>	<code>const SIZE = 4; x: array[1..2 * SIZE, 1..SIZE - 4] of Integer;</code>

5. ВАРІАНТИ ЗАДАЧ

1. Дано натуральне число n і дійсну квадратну матрицю $\mathbf{A} = (a_{ij})$ порядку n ($n \leq 10$). Одержати квадратну матрицю $\mathbf{B} = (b_{ij})$ порядку n , у якої для $i = 1, 2, \dots, n, j = 1, 2, \dots, n$

$$b_{ij} = \begin{cases} a_{ij} & \text{при } i \leq j, \\ -a_{ji} & \text{у супротивному випадку.} \end{cases}$$

2. Дано натуральне число n ($n \leq 10$) і дійсну квадратну матрицю порядку n . Сума елементів якої з діагоналей більше, – головної або побічної? Відповіддю повинне бути значення 1 для головної діагоналі і значення -1 для побічної діагоналі.
3. Дано натуральні числа m, n ($m, n \leq 10$) і дійсну матрицю розміру $m \times n$. Знайти суму найменших значень елементів її стовпців.
4. Дано натуральні числа m, n ($m, n \leq 10$) і цілочислову матрицю розміру $m \times n$. Змінити матрицю так, щоб у рядках залишилися елементи, які зустрічаються більше одного разу, а інші замінити нулями.
5. Дано натуральні числа m, n ($m, n \leq 10$) і цілочислову матрицю $\mathbf{A} = (a_{ij})_{mn}$. Замінити в матриці \mathbf{A} нулями елементи, що стоять у рядках або стовпцях, де є нулі. Дозволено використати один одновимірний допоміжний масив з m або n елементів.
6. Дано натуральні числа m, n ($m, n \leq 10$) і дійсну матрицю $\mathbf{A} = (a_{ij})_{mn}$. Замінити в матриці кожен елемент a_{ij} мінімальним з елементів підматриці, правий верхній кут якої збігається з елементом a_{ij} , а лівий нижній – з лівим нижнім кутом матриці \mathbf{A} .
7. Дано натуральні числа m, n ($m, n \leq 10$) і дійсну матрицю розміру $m \times n$. У кожному стовпці матриці знайти кількість елементів, менших середнього арифметичного всіх елементів цього стовпця.
8. Дано натуральні числа m, n ($m, n \leq 10$) і дійсну матрицю розміру $m \times n$. Визначити кількість її стовпців, всі елементи яких утворюють монотонні послідовності (монотонно зростальні або монотонно спадні).
9. Дано квадратний масив порядку n ($n \leq 10$). Знайти номери рядків, елементи кожного з яких утворюють симетричні послідовності (палиндроми).

10. Дано натуральні числа m, n ($m, n \leq 10$) і дійсну матрицю розміру $m \times n$. Визначити мінімальний серед елементів тих рядків, які впорядковані хоч за зростанням, хоч за убутанням.
11. Дано натуральні числа m, n ($m, n \leq 10$) і цілочислову матрицю розміру $m \times n$. Визначити номер останнього стовпця, у якому кількість додатних і від'ємних елементів однакова. Якщо такий стовпець відсутній, то результатом повинне бути число -1 .
12. Дано натуральні числа n, p, q ($n \leq 10, 1 \leq p \leq n, 1 \leq q \leq n$) і дійсну квадратну матрицю порядку n . Поміняти місцями одночасно p -й і q -й рядки і p -й і q -й стовпці матриці.
13. Дано натуральні числа m, n ($m, n \leq 10$) і двовимірний цілочисловий масив розміром розміру $m \times n$. Знайти номер останнього за порядком стовпця, у якому міститься найбільша кількість різних чисел.
14. Дано натуральне число n ($n \leq 10$) і дійсний квадратний масив розміру $n \times n$. Знайти значення індексів мінімального елемента з тих, що знаходяться у зафарбованій на рис. 5.1 частини масиву, включаючи фрагменти діагоналей.

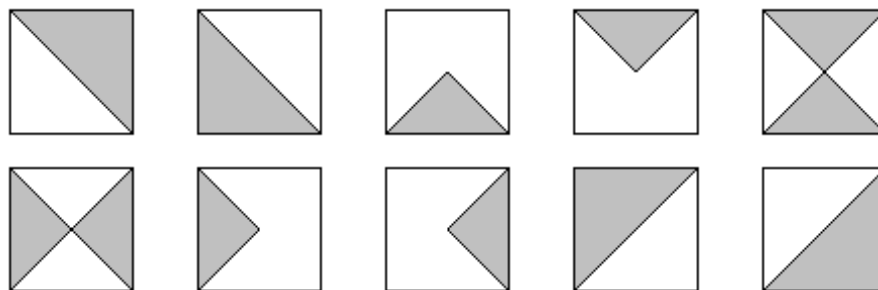


Рис. 5.1. Варіанти частин масивів

15. Дано натуральне число n ($n \leq 10$) і дійсний квадратний масив розміру $n \times n$. Обернути відносно вертикальної осі вміст зафарбованої на рис. 5.2 частини масиву, включаючи фрагменти діагоналей.

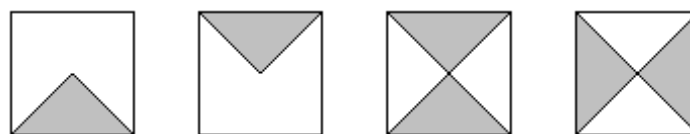


Рис. 5.2. Обертання відносно вертикальної осі

16. Дано натуральне число n ($n \leq 10$) і дійсний квадратний масив розміру $n \times n$. Обернути відносно вертикальної осі вміст зафарбованої на рис. 5.3 частини масиву, включаючи фрагменти діагоналей.

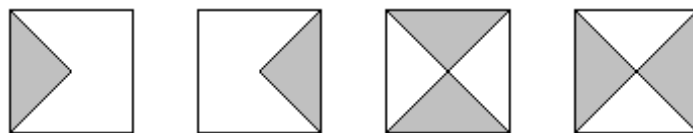


Рис. 5.3. Обертання відносно горизонтальної осі

17. У двовимірному масиві видалити рядок і стовпець, на перехресті яких знаходиться елемент, що є найближчим до середнього арифметичного усіх елементів.
18. У двовимірному масиві поміняти місцями рядки, що містять максимальний та мінімальний елементи.
19. Дано масив дійсних чисел розміру $n \times n$ ($n \leq 10$).
- а) Знайти суму елементів головної та побічної діагоналей.
 - б) Знайти найменше зі значень елементів побічної діагоналі та двох сусідніх з нею ліній.

СПИСОК ЛІТЕРАТУРИ

У разі використання мови C++:

1. Страуструп, Б. Язык программирования Си++ : Второе издание / Б. Страуструп. – К. : ДияСофт, 1993. – Ч. 1. – 264 с. ; Ч. 2. – 296 с.
2. Керниган, Б. Язык программирования Си / Б. Керниган, Д. Ритчи. – М. : Финансы и статистика, 1992. – 272 с.
3. Либерти, Джесс. Освой самостоятельно C++ за 21 день : учеб. пособ. / Джесс Либерти. – М. : Издательский дом «Вильямс», 2001. – 816 с.
4. Подбельский, В. В. Программирование на языке Си / В. В. Подбельский, С. С. Фомин. – М. : Финансы и статистика, 1999. – 600 с.
5. Подбельский, В. В. Язык Си++ / В. В. Подбельский. – М. : Финансы и статистика, 1999. – 560 с.
6. Савитч, Уолтер. Язык C++. Курс объектно-ориентированного программирования / Уолтер Савитч. – М. : Издательский дом «Вильямс», 2001. – 704 с.

У разі використання мови Delphi:

1. Безменов, М. І. Турбо Паскаль 7.0 : навч. посіб. / М. І. Безменов. – Х. : НТУ «ХПІ»; ПарусTM, 2005. – 240 с.
2. Кэнту, М. Delphi 7 : Для профессионалов / М. Кэнту – СПб. : Питер, 2004. – 1101 с.
3. Архангельский, А. Я. Программирование в Delphi 6 / А. Я. Архангельский. – М. : БИНОМ, 2002. – 1120 с.
4. Дарахвелидзе, П. Г. Программирование в Delphi 7 / П. Г. Дарахвелидзе, Е. П. Марков. – СПб. : БХВ-Петербург, 2003. – 784 с.
5. Культин, Н. Б. Основы программирования в Delphi 7 / Н. Б. Культин. – СПб.: БХВ-Петербург, 2003. – 608 с.
6. Пестриков, В. М. Delphi на примерах / В. М. Пестриков, А. Н. Маслобоев. – СПб. : БХВ-Петербург, 2005. – 496 с.
7. Ремкеев, А. А. Курс Delphi для начинающих. Полигон нестандартных задач / А. А. Ремкеев, С. В. Федотова. – М. : СОЛОН-Пресс, 2006. – 360 с.
8. Митчелл, К. Керман. Программирование и отладка в DelphiTM : учебный курс / Митчелл К. Керман. – М. : Вильямс, 2004. – 720 с.
9. Парижский, С. М. Delphi : Только практика / С. М. Парижский. – К. : МК-Пресс, 2005. – 208 с.
10. Культин, Н. Б. Основы программирования в Delphi 2007 / Н. Б. Культин. – СПб. : БХВ-Петербург, 2008. – 480 с.

Навчальне видання

Методичні вказівки
до лабораторної роботи
«Використання багатовимірних масивів»
з курсу «Програмування» для студентів напряму 6.040302 – Інформатика
і курсу «Програмування та алгоритмічні мови» для студентів напряму
6.040303 – Системний аналіз

Укладач БЕЗМЕНОВ Микола Іванович

Відповідальний за випуск О. С. Куценко
Роботу до видання рекомендував О. В. Горелий

За авторською редакцією

План 2010 р., поз. 16 / 46-10

Підп. до друку 15.03.2010 р. Формат $60 \times 84 \frac{1}{16}$. Папір офісний.
Riso-друк. Гарнітура Таймс. Ум. друк. арк. 1,3. Наклад 50 прим.
Зам. № 68. Ціна договірна.

Видавничий центр НТУ «ХП».
Свідоцтво про державну реєстрацію ДК № 3657 від 24.12.2009 р.
61002, Харків, вул. Фрунзе, 21

Друкарня НТУ «ХП», 61002, Харків, вул. Фрунзе, 21